

**IN THE CLAIMS:**

Please substitute the following claims for the same numbered claims in the application.

1. (Currently Amended) A method of providing access for a plurality of application-level users to an application comprising a plurality of resource class components comprising tiered layers of web servers, commerce servers, and database servers collectively executing on multiple networked machines, the method comprising:

receiving an incoming flow of requests from application-level users to use an application and components of said application;

providing, for each of the application-level users, respective sets of one or more application instances of each resource class component for the application on one or more machines, to service the incoming requests from respective application-level users to use the application;

directing each of the incoming requests to a particular application instance of an appropriate resource class component;

monitoring, for each of the application-level users, the number of request serviced by the application instances of the resource class components of the application;

identifying, within a time constraint, failures on any of said multiple networked machines;

changing the number of application instances of one or more resource class components in response to the monitored number of requests for each resource class component and based on machines comprising failures;

maintaining a record of the current rate of requests received from respective application-

level users, based on the monitored number of serviced requests; and

~~using hit weights corresponding to a number of hits allocated for each resource instance and an allocated weight for said each resource instance, collectively and automatically allocating fractions of different resource class components to a particular application-level user in response to the changed number of application instances of one or more resource class components by~~  
using a computational load of each request imposing on said application, wherein said computational load corresponds to a number of requests allocated for each resource instance, wherein said machines comprising failures are prevented from receiving allocations of resources.

2. (Previously Presented) The method as claimed in claim 1, further comprising directing each of the incoming requests from respective application-level users to a particular application instance of an appropriate resource class component from a respective set of one or more application instances of each resource class component, said particular application instance being identified as the least loaded of the application instances of the appropriate resource class component from that respective set.

3. (Previously Presented) The method as claimed in claim 1, wherein the step of providing application instances of each resource class component further comprises:

initiating one or more application instance of one or more resource class on a plurality of machines to service incoming requests to use the application; and

terminating one or more application instances of each resource class on a plurality of machines to service incoming requests to use the application.

4. (Previously Presented) The method as claimed in claim 1, wherein requests from application-level users to use the application are stored in a queue for execution by a particular application instance of the appropriate resource class on a first-in-first-out basis.
5. (Previously Presented) The method as claimed in claim 1, further comprising maintaining a record of service obligations to respective application-level users.
6. (Previously Presented) The method as claimed in claim 5, further comprising changing, for each of the application-level users, the number of application instances of each resource class component in response to the monitored number of requests for each resource class component, wherein the service obligations to respective application-level users are at least met.
7. (Cancelled).
8. (Previously Presented) The method as claimed in claim 1, wherein said step of changing the number of application instances of said one or more resource classes is (i) at least partly based upon said recorded current rate of requests received from respective application-level users, and (ii) at least partly based on predetermined information that correlates changes in request rates with changes in the corresponding number of application instances of said one or more resource classes required to service said request rates.
9. (Previously Presented) The method as claimed in claim 1, wherein one or more of the application-level users are organizations, and the requests are generated by individuals associated

09/921,868

with the respective organization.

10-12. (Canceled).

13. (Currently Amended) A method of providing access for a plurality of application-level users to an application comprising a plurality of resource class components comprising tiered layers of web servers, commerce servers, and database servers collectively executing on multiple networked machines, the method comprising:

receiving an incoming flow of requests from application-level users to use an application and components of said application;

providing, for each of the application-level users, respective sets of one or more application instances of each resource class component ~~for the application on one or more machines~~; to service the incoming requests from the application-level users to use the application;

monitoring, for each of the application-level users, the resources currently available and resources currently consumed by the requests serviced by application instances of the resource class components of the application;

identifying, within a time constraint, failures on any of said multiple networked machines;

maintaining a record of resources currently available to respective application-level users; and a record of resources currently consumed by respective application-level users; both records of said resources being maintained in respect of each of the one or more application instances of each resource class components; and

09/921,868

5

adjusting the respective numbers of said one or more application instances of each resource class component; and

~~using hit weights corresponding to a number of hits allocated for each resource instance and an allocated weight for said each resource instance, collectively and automatically allocating fractions of different resource class components to a particular application-level user in response to a fluctuating number of application instances of one or more resource class components, collectively and automatically allocating fractions of different resource class components to a particular application-level user in response to a fluctuating number of application instances of one or more resource class components by using a computational load of each request imposing on said application, wherein said computational load corresponds to a number of requests allocated for each resource instance, wherein said machines comprising failures are prevented from receiving allocations of resources, and~~

wherein said application instances of each resource class component are adjusted for each application-level user based (i) at least partly on said records of resources currently available and currently consumed by respective application-level users, and (ii) at least partly on predetermined information that estimates the number of each resource class components required to service requests for said application instances of the resource class components, and (iii) at least partly on machines comprising failures.

14. (Cancelled).

15. (Currently Amended) A system for providing access for a plurality of application-level users to an application comprising a plurality of resource class components comprising tiered

09/921,868

layers of web servers, commerce servers, and database servers collectively executing on multiple networked machines, the system comprising:

means for receiving an incoming flow of requests from application-level users to use an application and components of said application;

means for providing, for each of the application-level users, respective sets of one or more application instances of each resource class component ~~for the application on one or more machines~~, to service the incoming requests from respective application-level users to use the application;

means for directing each of the incoming requests to a particular application instance of an appropriate resource class component;

means for monitoring, for each of the application-level users, the number of requests serviced by the application instances of the resource class components of the application;

means for identifying, within a time constraint, failures on any of said multiple networked machines;

means for changing the number of application instances of one or more resource class components in response to the monitored number of requests for each resource class component and based on machines comprising failures;

means for maintaining a record of the current rate of requests received from respective application-level users, based on the monitored number of serviced requests; and

means for ~~collectively and automatically allocating fractions of different resource class components to a particular application level user in response to the changed number of application instances of one or more resource class components using hit weights corresponding to a number of hits allocated for each resource instance and an allocated weight for said each~~

~~resource instance, collectively and automatically allocating fractions of different resource class components to a particular application-level user in response to the changed number of application instances of one or more resource class components by using a computational load of each request imposing on said application, wherein said computational load corresponds to a number of requests allocated for each resource instance, wherein said machines comprising failures are prevented from receiving allocations of resources.~~

16. (Currently Amended) A computer software program, recorded on a medium and capable of execution by computing means able to interpret the computer software program, for providing access for a plurality of application-level users to an application comprising a plurality of resource class components comprising tiered layers of web servers, commerce servers, and database servers collectively executing on multiple networked machines, the computer software program comprising:

code means for receiving an incoming flow of requests from application-level users to use an application and components of said application;

code means for providing, for each of the application-level users, respective sets of one or more application instances of each resource class component ~~for the application on one or more machines,~~ to service the incoming requests from respective application-level users to use the application;

code means for directing each of the incoming requests to a particular application instance of an appropriate resource class component;

code means for monitoring, for each of the application-level users, the number of requests serviced by the application instances of the resource class components of the application; and

09/921,868

code means for identifying, within a time constraint, failures on any of said multiple networked machines;

code means for changing the number of application instances of one or more resource class components in response to the monitored number of requests for each resource class component and based on machines comprising failures;

code means for maintaining a record of the current rate of requests received from respective application-level users, based on the monitored number of serviced requests; and

~~code means for collectively and automatically allocating fractions of different resource class components to a particular application-level user in response to the changed number of application instances of one or more resource class components using hit weights corresponding to a number of hits allocated for each resource instance and an allocated weight for said each resource instance.~~ collectively and automatically allocating fractions of different resource class components to a particular application-level user in response to the changed number of application instances of one or more resource class components by using a computational load of each request imposing on said application, wherein said computational load corresponds to a number of requests allocated for each resource instance, wherein said machines comprising failures are prevented from receiving allocations of resources.

17-18. (Canceled).

19. (Currently Amended) A system for providing access for a plurality of application-level users to an application comprising a plurality of resource class components comprising tiered layers of web servers, commerce servers, and database servers collectively executing on multiple

09/921,868



networked machines, the system comprising:

means for receiving an incoming flow of requests from application-level users to use an application and components of said application;

means for providing, for each of the application-level users, respective sets of one or more application instances of each resource class component ~~for the application on one or more machines~~, to service the incoming requests from the application-level users to use the application;

means for monitoring, for each of the application-level users, the resources currently available and resources currently consumed by the requests serviced by application instances of the resource class components of the application;

means for identifying, within a time constraint, failures on any of said multiple networked machines;

means for maintaining a record of resources currently available to respective application-level users; and a record of resources currently consumed by respective application-level users; both records of said resources being maintained in respect of each of the one or more application instances of each resource class components;

means for adjusting the respective numbers of said one or more application instances of each resource class component; and

~~means for collectively and automatically allocating fractions of different resource class components to a particular application-level user in response to a fluctuating number of application instances of one or more resource class components using hit weights corresponding to a number of hits allocated for each resource instance and an allocated weight for said each resource instance;~~ collectively and automatically allocating fractions of different resource class

09/921;868

10

components to a particular application-level user in response to a fluctuating number of application instances of one or more resource class components by using a computational load of each request imposing on said application, wherein said computational load corresponds to a number of requests allocated for each resource instance, wherein said machines comprising failures are prevented from receiving allocations of resources, and

wherein said application instances of each resource class component are adjusted for each application-level user based (i) at least partly on said records of resources currently available and currently consumed by respective application-level users, ~~and~~ (ii) at least partly on predetermined information that estimates the number of each resource class components required to service requests for said application instances of the resource class components, and (iii) at least partly on machines comprising failures.

20. (Currently Amended) A computer software program recorded on a medium and able to be executed by computing means able to interpret the computer software program, for providing access for a plurality of application-level users to an application comprising a plurality of resource class components comprising tiered layers of web servers, commerce servers, and database servers collectively executing on multiple networked machines, the computer software program comprising:

code means for receiving an incoming flow of requests from application-level users to use an application and components of said application;

code means for providing, for each of the application-level users, respective sets of one or more application instances of each resource class component ~~for the application on one or more machines,~~ to service the incoming requests from the application-level users to use the

application;

code means for monitoring, for each of the application-level users, the resources currently available and resources currently consumed by the requests serviced by application instances of the resource class components of the application;

code means for identifying, within a time constraint, failures on any of said multiple networked machines;

code means for maintaining a record of resources currently available to respective application-level users; and a record of resources currently consumed by respective application-level users; both records of said resources being maintained in respect of each of the one or more application instances of each resource class components;

code means for adjusting the respective numbers of said one or more application instances of each resource class component in response to monitored number of requests for each resource class component and based on machines comprising failures; and

~~code means for collectively and automatically allocating fractions of different resource class components to a particular application-level user in response to a fluctuating number of application instances of one or more resource class components using hit weights corresponding to a number of hits allocated for each resource instance and an allocated weight for said each resource instance;~~ collectively and automatically allocating fractions of different resource class components to a particular application-level user in response to a fluctuating number of application instances of one or more resource class components by using a computational load of each request imposing on said application, wherein said computational load corresponds to a number of requests allocated for each resource instance, wherein said machines comprising failures are prevented from receiving allocations of resources, and

09/921,868

12

wherein said application instances of each resource class component are adjusted for each application-level user based (i) at least partly on said records of resources currently available and currently consumed by respective application-level users, and (ii) at least partly on predetermined information that estimates the number of each resource class components required to service requests for said application instances of the resource class components, and (iii) at least partly on machines comprising failures.